Tp 1 en langage C

Description de l'environnement Borland C++

Pour le travail pratique en C, il faut utiliser un compilateur et un éditeur facile à utiliser. A titre d'exemple je décris ici l'utilisation de l'environnement Borland C++ (Version 3.1). Ce programme nous offre une surface de programmation confortable et rapide. Tout autre compilateur qui permet la programmation selon le standard ANSI-C fait aussi bien l'affaire. Borland C++ est une implémentation complète du standard C++ AT&T (Version 2.1). Le compilateur Borland C++ est capable de produire du code C 'pur'. D'autre part, Borland C++ nous offre toute une série de bibliothèques, qui nous permettent d'exploiter les capacités du PC.

Les menus

- `FILE` gestion des fichiers

- `EDIT` commandes d'édition du texte

- `SEARCH` recherche d'un texte, d'une déclaration de fonction, de la position d'une erreur dans le programme

- `RUN` exécution d'un programme en entier ou par parties

- `COMPILE` traduction et/ou enchaînement *(link)* des programmes

- `DEBUG` détection d'erreurs en inspectant les données, en insérant des points d'observation *(watch)* ou en insérant des points d'arrêt *(breakpoints)*

- `PROJECT` gestion de projets

- `OPTIONS` changement et sauvetage des réglages par défaut:

  • des menus

  • des options du compilateur

  • de l'environnement

- `WINDOW` visualisation et disposition des différentes fenêtres *(Message, Output, Watch, User, Project)* sur l'écran

- `HELP` système d'aide

L'interface utilisateur:

L'environnement Borland C++ nous permet l'utilisation confortable de plusieurs fenêtres sur l'écran. L'interaction des données des différentes fenêtres a rendu la recherche d'erreurs très efficace.



La gestion de projets multi-fichiers:

En pratique, les programmes sont souvent subdivisés en plusieurs sous-programmes ou modules, qui peuvent être répartis dans plusieurs fichiers. L'environnement Borland C++ nous offre un gestionnaire *'Project Manager'* qui détecte automatiquement les fichiers qui doivent être recompilés et enchaînés après une modification. Dans nos applications, nous allons uniquement utiliser des fonctions définies dans le même fichier que notre programme principal ou prédéfinies dans une bibliothèque standard; ainsi, nous n'aurons pas besoin de l'option 'Project'.

Le système d'aide:

Borland C++ est accompagné d'un programme d'aide puissant qui est appelé directement à partir de l'éditeur. Le système d'aide peut donner des informations ponctuelles sur un sujet sélectionné ou nous laisser choisir un sujet à partir d'une liste alphabétique:

F1 Aide sur l'environnement Borland C++

Shift-F1 Index alphabétique des mots-clefs

Ctrl-F1 Aide sur le mot actif (à la position du curseur)

Utilisation des bibliothèques de fonctions :

La pratique en C exige l'utilisation de bibliothèques de fonctions. Ces bibliothèques sont disponibles dans leur forme précompilée (extension: `.LIB`). Pour pouvoir les utiliser, il faut inclure des fichiers en-tête (*header files* – extension `.H`) dans nos programmes. Ces fichiers contiennent des *'prototypes'* des fonctions définies dans les bibliothèques et créent un lien entre les fonctions précompilées et nos programmes.

#include

L'instruction ***#include*** insère les fichiers en-tête indiqués comme arguments dans le texte du programme au moment de la compilation.

Identification des fichiers

Lors de la programmation en Borland C, nous travaillons donc avec différents types de fichiers qui sont identifiés par leurs extensions:

`*.C` fichiers source

`*.OBJ` fichiers compilés (versions objet)

`*.EXE` fichiers compilés et liés (versions exécutables)

`*.LIB` bibliothèques de fonctions précompilées

`*.H` fichiers en-tête (*header files*)

Exemple

Nous avons écrit un programme qui fait appel à des fonctions mathématiques et des fonctions graphiques prédéfinies. Pour pouvoir utiliser ces fonctions, le programme a besoin des bibliothèques:

```
MATHS.LIB
GRAPHICS.LIB
```




Nous devons donc inclure les fichiers en-tête correspondants dans le code source de notre programme à l'aide des instructions:

```
#include <math.h>
#include <graphics.h>
```

Après la compilation, les fonctions precompilées des bibliothèques seront ajoutées à notre programme pour former une version exécutable du programme (voir schéma).

Schéma: Bibliothèques de fonctions et compilation

Fichiers en-tête

```
graphics.h
/* declarations */
```

```
math.h
/* declarations */
```

```
. . .
```

Programme principal

```
#include . . .
#include . . .
#include <graphics.h>
#include <math.h>

main (void)
{
. . .
}
```

Compiler

code compilé
de la fonction 'main'

Bibliothèques précompilées

graphics.lib

maths.lib

Linker

Programme exécutable

.EXE

Les fonctions de la bibliothèque <*math*>

Les fonctions suivantes sont prédéfinies dans la bibliothèque standard <*math*>. Pour pouvoir les utiliser, le programme doit contenir la ligne:

```
#include <math.h>
```

Type des données

Les arguments et les résultats des fonctions arithmétiques sont du type **double**.

Fonctions arithmétiques

COMMANDE C	*EXPLICATION*	*LANG. ALGORITHMIQUE*
`exp(X)`	fonction exponentielle	e^X
`log(X)`	logarithme naturel	`ln(X), X>0`
`log10(X)`	logarithme à base 10	$\log_{10}(X)$, X>0
`pow(X,Y)`	X exposant Y	X^Y
`sqrt(X)`	racine carrée de X	pour X>0
`fabs(X)`	valeur absolue de X	|X|
`floor(X)`	arrondir en moins	`int(X)`
`ceil(X)`	arrondir en plus	
`fmod(X,Y)`	reste rationnel de X/Y (même signe que X)	pour X différent de 0
`sin(X) cos(X) tan(X)`	sinus, cosinus, tangente de X	
`asin(X) acos(X) atan(X)`	arcsin(X), arccos(X), arctan(X)	
`sinh(X) cosh(X) tanh(X)`	sinus, cosinus, tangente hyperboliques de X	

Remarque avancée

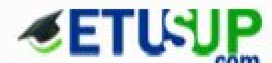
La liste des fonctions ne cite que les fonctions les plus courantes. Pour la liste complète et les constantes prédéfinies voir `<math.h>`.

Exercice 1 :

Essayez le programme suivant et modifiez-le de façon à ce qu'il affiche:

- **A^B,**
- **l'hypoténuse d'un triangle rectangle de côtés A et B,**
- **la tangente de A en n'utilisant que les fonctions sin et cos,**
- **la valeur arrondie (en moins) de A/B,**

```
#include <stdio.h>
main()
{
 double A;
 double B;
 double RES;
 /* Saisie de A et B */
 printf("Introduire la valeur pour A : ");
 scanf("%lf", &A);
 printf("Introduire la valeur pour B : ");
 scanf("%lf", &B);
 /* Calcul */
 RES = A*A;
```



```
  /* Affichage du résultat */
  printf("Le carré de A est %f \n", RES);
  /* Calcul */
  RES = B*B;
  /* Affichage du résultat */
  printf("Le carré de B est %f \n", RES);
  return 0;
}
```

Exercice 2 :

```
#include <stdio.h>
main()
{
   int N=10, P=5, Q=10, R;
   char C='S';

   N = 5; P = 2;
   Q = N++ > P || P++ != 3;
   printf ("C : N=%d P=%d Q=%d\n", N, P, Q);

   N = 5; P = 2;
   Q = N++ < P || P++ != 3;
   printf ("D : N=%d P=%d Q=%d\n", N, P, Q);

   N = 5; P = 2;
   Q = ++N == 3 && ++P == 3;
   printf ("E : N=%d P=%d Q=%d\n", N, P, Q);

   N=5; P=2;
   Q = ++N == 6 && ++P == 3;
   printf ("F : N=%d P=%d Q=%d\n", N, P, Q);

   N=C;
   printf ("G : %c %c\n", C, N);
   printf ("H : %d %d\n", C, N);
   return 0;
}
```

a) Sans utiliser l'ordinateur, trouvez et notez les résultats du programme ci-dessus.

b) Vérifiez vos résultats à l'aide de l'ordinateur.

Exercice 3 :

Considérez la séquence d'instructions suivante:

```
if (A>B) printf ("premier choix \n"); else
   if (A>10) printf ("deuxième choix \n");
   if (B<10) printf ("troisième choix \n");
   else printf ("quatrième choix \n");
```

a) Copiez la séquence d'instructions en utilisant des tabulateurs pour marquer les blocs **if - else** appartenant ensemble.



b) Déterminez les réponses du programme pour chacun des couples de nombres suivants et vérifiez à l'aide de l'ordinateur.

- A=10 et B=5
- A=5 et B=5
- A=5 et B=10 4ème choix
- A=10 et B=10 4 choix
- A=20 et B=10
- A=20 et B=20

Exercice 4 :

Considérez la séquence d'instructions suivante:

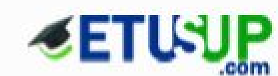
```
if (A>B)
   if (A>10)
   printf ("premier choix \n"); else if (B<10)
   printf ("deuxième choix \n"); else
   if (A==B) printf ("troisième choix \n");
   else printf ("quatrième choix \n");
```

a) Copiez la séquence d'instructions en utilisant des tabulateurs pour marquer les blocs if - else appartenant ensemble.

b) Pour quelles valeurs de A et B obtient-on les résultats:
premier choix, deuxième choix, ... sur l'écran? 3ème choix n'est jamais vérifié

c) Pour quelles valeurs de A et B n'obtient-on pas de réponse sur l'écran?

d) Notez vos réponses et choisissez vous-mêmes des valeurs pour A et B pour les vérifier à l'aide de l'ordinateur.




ETUSUP.com